# XRCE's Participation to Patent Mining Task at NTCIR-7

Stephane Clinchant Jean-Michel Renders
Xerox Research Centre Europe
6, Chemin de Maupertuis, 38240 Meylan, France
FirstName.LastName@xrce.xerox.com.

## Abstract

*In this first participation to the Patent Mining Task at NTCIR-7, our goal was to assess very simple large-scale categorization methods, especially in a cross-lingual framework. Our categorizers are instances of the "$k$-nearest neighbors" classifier. We used the Language Modelling approach to Information Retrieval as the building block to define similarity measures when computing the nearest neighbors. We also adopted a particular fusion scheme when building a class assignment function from the labels of nearest neighbors, that appears to be particularly efficient. As bilingual resources, we used simply a parallel part of the NTCIR-1 corpus, from which we extracted a bilingual lexicon. Even if this could look like a very crude approach, this bilingual lexicon, when integrated in a cross-lingual similarity measure, gave performance that is exactly at the same level as the monolingual case.*

**Keywords:** *NTCIR, kNN classifier, cross-lingual categorization.*

## 1 Introduction

This paper presents the XRCE's first participation to the Patent Mining Task. Patent documents raise some challenges to information systems. They are structured, large-scale data, with hierarchical and multilingual aspects. The large scale issue limits usage of complex machine learning techniques. Actually, we observed that most of the participants used variants of the $k$-nearest neighbours (kNN) methods (text retrieval models) in the dry-run, which requires low training time and complexity. Our approach is not different : it is also based on kNN classifiers, but using a similarity measure based on the Language Modeling approach to information retrieval, and adopting a special fusion scheme to combine the "votes" (labels) of the neighbors. In the next section, we will briefly explain the data and preprocessing we did to conduct the submitted runs. As we adopted a kNN-like classifier, the role of the metrics to compute textual similarities is particularly important; hence, the textual similarity measure that we used and that is based upon the language modeling approach to information retrieval [3] will be introduced. The decision function of the kNN classifier will also be described. Finally, we will show the performance of the official runs, before concluding on future works.

## 2 Data and Preprocessing

In this section, we describe the data, the tools and the preprocessing steps that we adopted for the patent hierarchical categorization task.

We participated in the monolingual English subtask, as well as in the cross-lingual Japanese to English subtask. Although our main goal was to deal with Japanese documents, we did not submit any run concerning Japanese patents due to resource constraints (lack of adequate Natural Language Processing tools for this language).

The English data consisted of two corpora: the PAJ and USPTO corpora. These corpora were simply tokenised (by a standard tokenizer) and indexed with the Lemur system [2], separately and without any stemming operation. We carried some preliminary categorization experiments using the dry-run test data in order to determine (1) which fields of the patent records have to be indexed, and (2) whether it is better to use only one corpus (PAJ or USPTO) or a combination of both corpora as indexed training documents. First, it appeared that USPTO results were better if we indexed only the title and abstract of the document (thus removing all claims and most of the text of a patent). Secondly, it turned out that using only the PAJ corpus gave much better results than using a combination of USPTO and PAJ corpora. One reason of this phenomenon is the possible bias between test queries and training documents (due to the construction mechanism of the dry-run test data, we understood from the organizers that the each scientific paper in the test set had a "patent" equivalent in the PAJ corpus, but not necessarily in the USPTO corpus). A second reason may rely in the fact that USPTO patents have a single label, while PAJ patents have multiple labels, which

better reflects the fact that patents are actually multi-topical.

Concerning the cross-lingual task, that basically needs a bilingual lexicon, we used the NTCIR-1 data to extract an aligned corpus of approximatively 300,000 titles. We used the *chasen* [1] tool to segment the Japanese texts. From this parallel corpus, a bilingual (Japanese-English) probabilistic dictionary was obtained using the *Giza++* [2] word alignment algorithm.

To sum up our approach, we performed a naive and simple preprocessing of the data: indexing the raw data, without taking into account structure in document or anything else. The only selection or filtering was to discard the USPTO corpus from the training set.

## 3 Text Similarity Measure

As our text categorization method needs some efficient textual similarity measure to determine the neighbours of a test document, we describe here the one we adopted for our kNN classifier. This measure comes from state-of-the-art information retrieval techniques: namely, the language modeling approach to information retrieval. Let us first consider the test document as a query, and the patent corpus as a target corpus where we have to rank documents according to their relevance (similarity) to the query.

The core idea of language models is to determine the probability $P(q|d)$ — the probability that the query would be generated from a particular document. Formally, given a query $q$, the language model approach to IR [3] scores documents $d$ by estimating $P(q|d)$, the probability of the query according to some language model of the document. Using some independence assumption, for a query $q = \{q_1, \ldots q_\ell\}$, we get:

$$P(q|d) = \prod_{i=1}^{\ell} P(q_i|d). \quad (1)$$

We assume that for each document there exists some probability distribution over words – a Language Model – parametrized by a vector $\theta_d$. Abusively we note $P(q|d) \equiv P(q|\theta_d)$. Standard language models in information retrieval are multinomial distributions: the language model of a document is defined by its parameter vector $\theta_d$, whose dimension is the size of the vocabulary. As this multinomial parameter is normalized (the sum of its components sums up to one), another notation is used : $\theta_{dw} = P(w|d)$.

For each document $d$, a simple language model could be obtained by considering the frequency of words in $d$, $P_{ML}(w|d) \propto \#(w,d)$ (this is the Maximum Likelihood, or ML, estimator). The probabilities can more conveniently be smoothed by the corpus language model $P_{ML}(w|\mathcal{C}) \propto \sum_d \#(w,d)$. The resulting language model is:

$$P(w|d) = \lambda P_{ML}(w|d) + (1-\lambda) P_{ML}(w|\mathcal{C}). \quad (2)$$

The reasons of smoothing are twofold: first a word can be present in a query but absent in a document. However this fact could not make it impossible that the query was produced by the LM of the document and the document should give it a probability. The second reason is to play a role like the Inverse Document Frequency. Smoothing enables implicitly to renormalize the frequency of one word in a document with respect to its occurrence in the corpus. Others smoothing methods could be applied (Dirichlet smoothing, Absolute Discounting, ...) and can be found in [4]. The *Query Likelihood* approach above gives an intuitive view of how language models works in information retrieval. Other equivalent ranking functions can be considered and lead to the same ranking function as the *Query Likelihood* formulation. For example the *KL-divergence* and the *Cross-Entropy* functions can also be used in information retrieval. Let $\theta_q$ be a multinomial parameter for the language model of a query $q$, $\theta_d$ the language model for a document $d$, the cross-entropy function between these two objects is:

$$CE(\theta_q|\theta_d) = \sum_w P(w|q) \log(P(w|d)) = \sum_w \theta_{qw} \log(\theta_{dw}) \quad (3)$$

Note that, for the cross-lingual categorization task, we need to define cross-lingual textual similarity too. As far as cross-lingual IR is concerned, the core idea remains the same: modeling the probability of the query given the document. Let $q_s$ be the query in some source language, $w_s$ a word in the source language, $d_t$ a document in the target language, $w_t$ a word in the target language, $P(w_t|w_s)$ the probability that word $w_s$ is translated into $w_t$.

The method we adopted consists then is translating the query (test document) into a target query model [1]. Afterwards, a monolingual search is performed, using a ranking criterion such as the Cross-Entropy:

$$\begin{aligned} CE(q_s|d_t) &= \sum_{w_t} P(w_t|q_s) \log P(w_t|d_t) \\ &= \sum_{w_t,w_s} P(w_t|w_s,q_s) P(w_s|q_s) \log P(w_t|d_t) \\ &\cong \sum_{w_t,w_s} P(w_t|w_s) P(w_s|q_s) \log P(w_t|d_t) \end{aligned} \quad (4)$$

[1] http://chasen.aist-nara.ac.jp/chasen/distribution.html
[2] http://www.fjoch.com/GIZA++.html

## 4 Label Fusion

Traditional kNN classifiers first compute distances between a test document and documents of the target training corpus (what was explained in the previous section) and secondly apply some rules for the class assignment. The case of the PAJ corpus differs from more traditional scenarios: patents are multilabelled and labels are hierarchical. The two features may require new rules or decisions in the second step of a kNN-classifier.

Given a test document $q$ and a ranked list of scores for the top $N$ (most similar) documents, denoted as $N(q)$, we describe now our ranking strategy. Here $N$ is typically chosen as 100 but, as we will see, the method is relatively robust to this choice as the contribution of any document is weighted by its relative similarity with respect to the query $q$. For each class $c$ (a class, or label, is any leave of the IPC taxonomy), we compute the following score:

$$s(c|q) = [\sum_{d \in N(q) \wedge labels(d) \ni c} \widetilde{sim}(d,q)]^{(1-\alpha)} .[\max_{d \in N(q) \wedge labels(d) \ni c} \widetilde{sim}(d,q)]^{\alpha} \quad (5)$$

where $labels(d)$ is the set of labels associated to training document $d$ and $\widetilde{sim}(d,q)$ is the "max-min"-normalized similarity value between $d$ and $q$ obtained from the Cross-Entropy measure as defined by equations 3 or 4 (the minimum similarity is mapped to 0, while the max value is mapped to 1).

Basically, this fusion scheme is some generalized harmonic mean between two kinds of fusion operators, the first being the sum (or average) operator, while the other one is the disjunction (max) operator. During the preliminary experimental phase, we found that setting the $\alpha$ parameter to value close to 0.15 gave good results, at least for the dry-run test data. As the class score could be considered as an image of the probability that test document $q$ belongs to class $c$, we finally rank the classes (labels) by decreasing value of their score, to constitute the output of our system .

We also tried to integrate the hierarchy information in some way, by propagating class scores to their "brothers" (or, more generally, to other classes with a diffusion weight that decreases with their mutual distances in the tree), but we did not manage to discover any way to improve significantly the categorization performance with respect to the flat version.

## 5 Official Runs

Our official monolingual English runs consist of a nearest neighbor search with language models followed by applying the fusion algorithm we proposed. We selected different values for the smoothing parameter ($\lambda$) in the language model and the combination parameter ($\alpha$) in the fusion equation 5, and these different variants gave all the runs we proposed. In fact, the performance of runs are close to each other. Although our approach is simple (almost no preprocessing, state-of-the-art textual similarity), our team ranked second in the official run. The average of our 4 runs is around $42\%$ mean average precision.

The cross-lingual runs were obtained by translating the queries with the bilingual dictionary we extracted from a parallel part of the NTCIR-1 corpus, and then applying the same process as in the monolingual runs. There were only two teams participating in this challenge. We ranked first, with a performance that is comparable with the monolingual runs. However, it is hard to draw conclusions and compare approaches when there are not enough participants. Our single cross-lingual run achieved a $43\%$ mean average precision.

## 6 Conclusion and Future Works

We have presented our approach to the Patent Mining Task. Our approach was quite simple and used state-of-the-art techniques in information retrieval, which gave relatively good performance. As our initial goal was to deal with Japanese documents, further work will evaluate the approach on the monolingual Japanese subtask. We plan also to investigate more rigorously the hierarchical case, in order to exploit the taxonomy information to improve categorization results. Finally, we want to design specific approaches to overcome the issue raised by the differences in style, structure and expression between the test and the training corpora.

## References

[1] W. Kraaij, J.-Y. Nie, and M. Simard. Embedding web-based statistical translation models in cross-language information retrieval. *Comput. Linguist.*, 29(3):381–419, 2003.

[2] Lemur. http://www.lemurproject.org/.

[3] J. Ponte and W. Croft. A language modelling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281. ACM, 1998.

[4] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc to information retrieval. In *Proceedings of SIGIR'01*, pages 334–342. ACM, 2001.